



EXPRESS MAIL NO. EV336611585US

---

# BREVET D'INVENTION

---

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 03 AVR. 2003

Pour le Directeur général de l'Institut  
national de la propriété industrielle  
Le Chef du Département des brevets

Martine PLANCHE

INSTITUT  
NATIONAL DE  
LA PROPRIÉTÉ  
INDUSTRIELLE

SIEGE  
26 bis, rue de Saint Petersburg  
75800 PARIS cedex 08  
Téléphone : 33 (0)1 53 04 53 04  
Télécopie : 33 (0)1 53 04 45 23  
[www.inpi.fr](http://www.inpi.fr)



100

**THIS PAGE BLANK (USPTO)**



26 bis, rue de Saint Pétersbourg  
75800 Paris Cedex 08  
Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

# BREVET D'INVENTION CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle - Livre VI



N° 11354\*03

## REQUÊTE EN DÉLIVRANCE page 1/2



Cet imprimé est à remplir lisiblement à l'encre noire

DB 540 • B / 210502

<b>REMISE DES PIÈCES</b> <b>DATE</b> 17 FEV 2003 <b>LIEU</b> 13 INPI MARSEILLE <b>N° D'ENREGISTREMENT</b> 0301879 <b>NATIONAL ATTRIBUÉ PAR L'INPI</b> <b>DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI</b> 17 FEV. 2003		<b>1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE</b> OMNIPAT MARCHAND André 24 Place des Martyrs de la Résistance 13100 AIX EN PROVENCE	
<b>Vos références pour ce dossier (facultatif)</b> 100213 FR			
<b>Confirmation d'un dépôt par télécopie</b>		<input type="checkbox"/> N° attribué par l'INPI à la télécopie	
<b>2 NATURE DE LA DEMANDE</b>		<b>Cochez l'une des 4 cases suivantes</b>	
Demande de brevet		<input checked="" type="checkbox"/>	
Demande de certificat d'utilité		<input type="checkbox"/>	
Demande divisionnaire		<input type="checkbox"/>	
<i>Demande de brevet initiale</i> <i>ou demande de certificat d'utilité initiale</i>		N° _____ Date _____ N° _____ Date _____	
Transformation d'une demande de brevet européen <i>Demande de brevet initiale</i>		<input type="checkbox"/> N° _____ Date _____	
<b>3 TITRE DE L'INVENTION (200 caractères ou espaces maximum)</b> Procédé de gestion d'une pile de microprocesseur pour la sauvegarde de données contextuelles			
<b>4 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE</b>		Pays ou organisation _____ N° _____ Date _____ Pays ou organisation _____ N° _____ Date _____ Pays ou organisation _____ N° _____ <input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»	
<b>5 DEMANDEUR (Cochez l'une des 2 cases)</b>		<input checked="" type="checkbox"/> Personne morale <input type="checkbox"/> Personne physique	
Nom ou dénomination sociale		STMICROELECTRONICS SA	
Prénoms			
Forme juridique		Société Anonyme	
N° SIREN		3 4 1 4 5 9 3 8 6	
Code APE-NAF		3 2 1 B	
Domicile ou siège	Rue	29 Boulevard Romain Rolland	
	Code postal et ville	9 2 1 2 0 MONTRouGE	
	Pays	FRANCE	
Nationalité		FRANCE	
N° de téléphone (facultatif)		N° de télécopie (facultatif)	
Adresse électronique (facultatif)			
<input type="checkbox"/> S'il y a plus d'un demandeur, cochez la case et utilisez l'imprimé «Suite»			



# BREVET D'INVENTION

## CERTIFICAT D'UTILITÉ

REQUÊTE EN DÉLIVRANCE  
page 2/2

**BR2**

REMISE DES PIÈCES DATE <b>17 FEV 2003</b> LIEU <b>13 INPI MARSEILLE</b> N° D'ENREGISTREMENT <b>0301879</b> NATIONAL ATTRIBUÉ PAR L'INPI		Réservé à l'INPI	DB 540 W / 210502
<b>6 MANDATAIRE</b>			
Nom		MARCHAND	
Prénom		André	
Cabinet ou Société		OMNIPAT	
N° de pouvoir permanent et/ou de lien contractuel			
Adresse	Rue	24 Place des Martyrs de la Résistance	
	Code postal et ville	13 10 10 AIX EN PROVENCE	
	Pays	FRANCE	
N° de téléphone (facultatif)		04.42.99.06.60	
N° de télécopie (facultatif)		04.42.99.06.69	
Adresse électronique (facultatif)			
<b>7 INVENTEUR (S)</b>		Les inventeurs sont nécessairement des personnes physiques	
Les demandeurs et les inventeurs sont les mêmes personnes		<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non : Dans ce cas remplir le formulaire de Désignation d'inventeur(s)	
<b>8 RAPPORT DE RECHERCHE</b>		Uniquement pour une demande de brevet (y compris division et transformation)	
Établissement immédiat ou établissement différé		<input checked="" type="checkbox"/> Établissement immédiat <input type="checkbox"/> Établissement différé	
Paiement échelonné de la redevance (en deux versements)		Uniquement pour les personnes physiques effectuant elles-mêmes leur propre dépôt <input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non	
<b>9 RÉDUCTION DU TAUX DES REDEVANCES</b>		Uniquement pour les personnes physiques <input type="checkbox"/> Requête pour la première fois pour cette invention (joindre un avis de non-imposition) <input type="checkbox"/> Obtenue antérieurement à ce dépôt pour cette invention (joindre une copie de la décision d'admission à l'assistance gratuite ou indiquer sa référence): AG [ ] [ ] [ ] [ ]	
<b>10 SÉQUENCES DE NUCLEOTIDES ET/OU D'ACIDES AMINÉS</b>		<input type="checkbox"/> Cochez la case si la description contient une liste de séquences	
Le support électronique de données est joint		<input type="checkbox"/> Oui <input type="checkbox"/> Non	
La déclaration de conformité de la liste de séquences sur support papier avec le support électronique de données est jointe		<input type="checkbox"/> Oui <input type="checkbox"/> Non	
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes			
<b>11 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE</b> (Nom et qualité du signataire)		<b>VISA DE LA PRÉFECTURE OU DE L'INPI</b> 	
MARCHAND André - CPI N° 95 0303 OMNIPAT			

PROCEDE DE GESTION D'UNE PILE DE MICROPROCESSEUR POUR LA  
SAUVEGARDE DE DONNEES CONTEXTUELLES

La présente invention concerne les microprocesseurs, et plus particulièrement un procédé de gestion de la pile d'un microprocesseur pour la sauvegarde de données contextuelles lors d'un  
5 basculement d'un premier à un second programme.

La présente invention vise en particulier la réalisation d'un microprocesseur à plan mémoire étendu qui soit compatible avec un microprocesseur à plan mémoire réduit.

10 La figure 1 représente de façon très schématique un microprocesseur 8 bits de première génération MPI commercialisé par la demanderesse. Le microprocesseur comprend une unité centrale ou CPU et un plan mémoire MEM1 adressable par l'intermédiaire d'un bus d'adresse  
15 ADBUS de 16 bits et accessible par l'intermédiaire d'un bus de données DBUS de 8 bits. Le plan mémoire présente ainsi une taille maximale de 64 Koctets ( $2^{16}$  octets) et comprend diverses zones mémoire parmi lesquelles on distingue notamment une zone mémoire volatile DMEM, pour  
20 enregistrer des données d'application, une zone mémoire volatile formant une pile STK ("Stack"), ainsi qu'une zone mémoire programme PRGMEM non volatile pour enregistrer un ou plusieurs programmes application. Les zones mémoire volatiles sont par exemple de type RAM et  
25 les zones mémoire non volatiles de type ROM, EPROM, EEPROM ou Flash.

Le CPU comprend divers registres de 8 bits dans lesquels sont stockées des données contextuelles CTX. On distingue un registre de code condition CCR, un accumulateur ACC, un registre d'index X, et des registres PCL, PCH recevant respectivement l'octet de poids faible et l'octet de poids fort d'un compteur ordinal PC ("program counter"). Pendant l'exécution d'un programme, le compteur ordinal PC désigne l'adresse de l'instruction suivante à exécuter et est incrémenté après lecture de l'instruction précédente. Le registre d'index X contient des données nécessaires à l'exécution d'instructions avec adressage indexé ou d'opérations prévues par un programme application. L'accumulateur ACC contient le résultat de calculs ou d'opérations effectuées par l'unité arithmétique et logique du CPU (non représentée).

Comme représenté en figure 1A, le registre CCR contient des drapeaux (flags) indiquant des résultats d'opérations ou d'instructions, typiquement les drapeaux C (Carry), Z (Zéro), N (Négatif), H (Half Carry), IO et I1 (masques d'interruption) et contient ici deux emplacements (deux bits) non utilisés.

Lorsque le CPU bascule d'un programme en cours d'exécution à un autre programme, généralement un sous-programme d'interruption, le CPU sauvegarde dans la pile STK les données contextuelles CTX présentes dans les registres CCR, ACC, X, PCH, PCL puis lit l'adresse de la première instruction du sous-programme à un emplacement de la mémoire programme PRGMEM désigné par un vecteur d'interruption, charge cette nouvelle adresse dans le compteur ordinal PC, dont le contenu initial a été sauvegardé dans la pile, et exécute le sous-programme.

La sauvegarde des données contextuelles CTX est effectuée octet par octet dans la pile STK, à partir d'une adresse basse ADL et jusqu'à une adresse haute ADH qui sont invariables et fixées par le constructeur. L'adresse de sauvegarde des données contextuelles est

indiquée par un pointeur de pile SP ("Stack Pointer") stocké dans des registres SPL, SPH du CPU, ces registres contenant respectivement l'octet de poids faible et l'octet de poids fort du pointeur de pile. Après chaque  
5 sauvegarde d'un octet de données contextuelles, le pointeur de pile est incrémenté d'une unité.

La taille de la pile et le nombre d'octets de données contextuelles à sauvegarder à chaque basculement d'un programme à un autre, détermine le nombre de  
10 basculements dans des programmes imbriqués pouvant être effectués par le CPU, soit le nombre d'interruptions en latence pouvant être traitées successivement de façon cumulative. Si les données contextuelles comprennent cinq octets comme indiqué plus haut, et si la pile  
15 s'étend par exemple sur 25 lignes du plan mémoire MEM1, le CPU peut effectuer 25/5 soit 5 basculements en cascade sans débordement de la pile. On trouve ainsi dans la pile des données contextuelles empilées CTX1, CTX2, CTX3, ... CTXj comme représenté en figure 1.

20 A chaque retour à un programme initial (qui peut être un sous-programme d'interruption dans le cas de sous-programmes imbriqués) les données contextuelles mémorisées dans la pile STK sont récupérées par le CPU à l'emplacement indiqué par le pointeur de pile SP, l'état  
25 des registres CCR, ACC, X, PCH, PCL est restauré et le CPU reprend l'exécution du programme initial à l'endroit où il avait été interrompu. A chaque restauration d'un octet de données contextuelles, le pointeur de pile est décrémenté d'une unité, selon le principe  
30 d'empilement/dépilage LIFO ("Last In First Out") dans lequel la dernière donnée entrée est la première donnée lue.

La figure 2 représente schématiquement un microprocesseur 8 bits MP2 de nouvelle génération, visé  
35 par la demanderesse. Le microprocesseur MP2 se distingue du microprocesseur MP1 par le fait qu'il comprend un plan mémoire étendu MEM2 adressable sous 24 bits au lieu

de 16 bits, via le bus d'adresse ADBUS. Ainsi, le plan mémoire MEM2 comprend ici 256 secteurs SCT0, SCT1, ... SCT255 de 64 KO chacun, et le premier secteur SCT0 correspond au plan mémoire MEM1 du microprocesseur MP1.

Cette extension du plan mémoire nécessite la prévision dans le CPU d'un registre supplémentaire PCE ("Program Counter Extended Address"), pour obtenir un compteur ordinal PC de 24 bits au lieu de 16, adapté à la taille du plan mémoire étendu.

A chaque basculement d'un premier à un second programme, le CPU doit alors sauvegarder dans la pile STK le contenu des six registres CCR, ACC, X, PCE, PCH, PCL, soit six octets de données contextuelles CTX au lieu de cinq.

Il est souhaité que le microprocesseur MP2 soit compatible avec le microprocesseur MP1, c'est-à-dire qu'il puisse recevoir et exécuter des programmes application développés pour le microprocesseur MP1. L'espace alloué à la pile STK doit ainsi rester constant est reste délimité par les adresses ADL, ADH dans le secteur SCT0. Dans le cas contraire, si un changement de la taille et/ou de l'emplacement de la pile du microprocesseur était prévu, un tel changement pourrait conduire à un empiètement des données de programme et des données d'application du programme conçu pour le microprocesseur MP1, sur l'emplacement de la pile du microprocesseur MP2.

Toutefois Le problème suivant apparaît : la taille de la pile STK étant maintenue constante, la prévision du registre d'adressage étendu PCE limite le nombre de programmes pouvant être imbriqués puisqu'un octet de données contextuelles supplémentaire doit être sauvegardé. En reprenant l'exemple d'une pile comprenant 25 emplacements de sauvegarde, le nombre de programmes pouvant être imbriqués est maintenant de 25/6 soit 4 au lieu de 5 (une ligne de la pile STK restant inutilisée).



Ceci constitue un obstacle à la compatibilité visée, car un programme développé pour l'ancienne génération de microprocesseurs pourrait exploiter au maximum les ressources de la pile. L'exécution d'un tel  
5 programme va se traduire par l'apparition d'un dysfonctionnement puisque le CPU ne pourra pas incrémenter le pointeur de pile au-delà de l'adresse ADH, de sorte que des données contextuelles vont être perdues.

10 La présente invention vise, notamment, à pallier cet inconvénient.

Plus particulièrement, un objectif de l'invention est de préserver la compatibilité entre deux générations de microprocesseurs se distinguant par une extension du  
15 plan mémoire et une extension de la taille du compteur ordinal.

Un objectif plus général de la présente invention, est de prévoir un procédé de sauvegarde de données contextuelles permettant d'optimiser la gestion de  
20 l'espace alloué à la pile afin de maintenir un degré élevé d'imbrication de programmes.

Ces objectifs sont atteints par la prévision d'un microprocesseur comprenant une unité centrale et un plan mémoire, l'unité centrale comprenant des registres  
25 contenant des données contextuelles et un pointeur de pile, le plan mémoire comprenant une pile pour la sauvegarde de données contextuelles, l'unité centrale étant agencée pour sauvegarder des données contextuelles lors d'un basculement d'un premier à un second  
30 programme, avec la particularité que l'unité centrale est agencée pour sauvegarder des données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur d'au moins un drapeau stocké dans un registre à sauvegarder.

35 Selon un mode de réalisation, l'unité centrale est agencée pour modifier la valeur du drapeau en fonction du contenu d'un registre, avant de sauvegarder des

données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur du drapeau.

Selon un mode de réalisation, l'unité centrale est agencée pour modifier la valeur du drapeau en fonction  
5 du contenu d'un registre d'adressage étendu d'un compteur ordinal de l'unité centrale.

Selon un mode de réalisation, l'unité centrale est agencée pour, quand le contenu du registre d'adressage étendu est égal à 0, sauvegarder tous les registres de  
10 l'unité centrale contenant des données contextuelles à l'exception du registre d'adressage étendu, ou quand le contenu du registre d'adressage étendu n'est pas égal à 0, sauvegarder tous les registres de l'unité centrale contenant des données contextuelles y compris le  
15 registre d'adressage étendu.

Selon un mode de réalisation, l'unité centrale est agencée pour effectuer un test sur la valeur du drapeau afin de déterminer le nombre de registres à sauvegarder.

Selon un mode de réalisation, l'unité centrale est  
20 agencée pour, lors du retour au premier programme, restaurer le registre contenant le drapeau et restaurer des données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur du drapeau présent dans le registre restauré.

25 Selon un mode de réalisation, l'unité centrale est agencée pour sauvegarder en dernier le registre contenant le drapeau.

Selon un mode de réalisation, le drapeau comprend au moins un bit d'un registre contenant des drapeaux de  
30 code condition.

L'invention concerne également un procédé de gestion de la pile d'un microprocesseur comprenant une unité centrale et un plan mémoire, l'unité centrale comprenant des registres contenant des données  
35 contextuelles et un pointeur de pile, la pile étant une zone du plan mémoire dédiée à la sauvegarde de données contextuelles lors d'un basculement d'un premier à un

second programme, le procédé comprenant la sauvegarde de données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur d'au moins un drapeau stocké dans un registre à sauvegarder.

5        Selon un mode de réalisation, le procédé comprend une étape de modification de la valeur du drapeau en fonction du contenu d'un registre, avant la sauvegarde de données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur du  
10 drapeau.

      Selon un mode de procédé, la valeur du drapeau est modifiée en fonction du contenu d'un registre d'adressage étendu d'un compteur ordinal de l'unité centrale.

15        Selon un mode de réalisation, le procédé comprend des étapes dans lesquelles, quand le contenu du registre d'adressage étendu est égal à 0, sauvegarder tous les registres de l'unité centrale contenant des données contextuelles à l'exception du registre d'adressage  
20 étendu, ou quand le contenu du registre d'adressage étendu n'est pas égal à 0, sauvegarder tous les registres de l'unité centrale contenant des données contextuelles y compris le registre d'adressage étendu.

      Selon un mode de réalisation, le procédé comprend  
25 une étape de test de la valeur du drapeau pour déterminer le nombre de registres contenant les données à sauvegarder.

      Selon un mode de réalisation, le procédé comprend des étapes de restauration du registre contenant le  
30 drapeau, puis de restauration de données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur du drapeau présent dans le registre restauré.

      Selon un mode de réalisation, le registre contenant  
35 le drapeau est sauvegardé le dernier et est restauré le premier.

Selon un mode de réalisation, le drapeau est formé par au moins un bit d'un registre contenant des drapeaux de code condition.

Ces objets, caractéristiques et avantages, ainsi  
5 que d'autres de la présente invention, seront exposés plus en détail dans la description ci-après de deux modes de réalisation d'un microprocesseur selon l'invention, faite à titre non limitatif en relation avec les figures jointes parmi lesquelles :

10 - la figure 1 précédemment décrite représente sous forme de bloc l'architecture d'un microprocesseur classique,  
- la figure 1A représente le contenu d'un registre du microprocesseur de la figure 1,  
- la figure 2 représente sous forme de bloc  
15 l'architecture d'un microprocesseur selon l'invention, ayant un plan mémoire étendu,  
- la figure 2A représente le contenu d'un registre du microprocesseur de la figure 2,  
- la figure 3 est un organigramme représentant des  
20 étapes d'empilement de données selon un procédé de gestion de pile selon l'invention,  
- la figure 4 est un organigramme représentant des étapes de dépilement de données selon le procédé de gestion de pile selon l'invention,  
25 - la figure 5 représente l'architecture d'un microprocesseur comprenant un nombre quelconque de données contextuelles, et illustre une généralisation du procédé de l'invention à un tel microprocesseur,  
- la figure 6 est un organigramme représentant des  
30 étapes d'empilement de données selon une variante du procédé de gestion de pile selon l'invention, et,  
- la figure 7 est un organigramme représentant des étapes de dépilement de données selon la variante du procédé de gestion de pile selon l'invention.

La figure 2 déjà décrite représente schématiquement l'architecture d'un microprocesseur à plan mémoire étendu MP2 dans lequel est mis en œuvre un procédé de gestion de pile STK selon l'invention.

5 Comme indiqué précédemment, le compteur ordinal PC comprend ici 24 bits stockés en trois octets dans des registres PCE, PCH, PCL du CPU. Le plan mémoire étendu MEM2 comporte 256 secteurs SCT0 à SCT255 de 64 Koctets chacun.

10 Le CPU est agencé pour exécuter, en sus d'un jeu d'instructions de nouvelle génération permettant de travailler sur toute l'étendue du plan mémoire, un jeu d'instructions simplifiées permettant de travailler sur le premier secteur SCT0, lequel conserve la même  
15 structure que le plan mémoire MEM1 du microprocesseur MP1 de première génération représenté en figure 1, afin que les deux générations de microprocesseurs soient compatibles.

Lors de l'exécution de ces instructions  
20 simplifiées, le registre d'adressage étendu PCE du compteur ordinal PC est remis à zéro et est laissé à zéro, puisque les instructions des programmes d'ancienne génération sont enregistrées dans le premier secteur SCT0, dont les adresses sont codées uniquement sur 16  
25 bits et peuvent être gérées au moyen des registres PCH, PCL. Le nouveau jeu d'instructions est par contre destiné à des programmes enregistrés dans tous les secteurs SCT1 à SCT255 et utilise le registre PCE.

Le CPU, qui est classiquement une machine d'état à  
30 logique câblée ou une machine logique microprogrammée, est conçu pour sauvegarder les données contextuelles présentes dans les registres CCR, ACC, X, PCH, PCL, CPE conformément à un procédé de gestion de pile selon l'invention.

35 Selon ce procédé, le CPU ne sauvegarde pas le contenu du registre d'adressage étendu PCE lorsque celui-ci est nul. De plus, une information indiquant que

le registre d'adressage étendu PCE a été ou non sauvegardé est insérée dans les données contextuelles afin que le CPU sache, lors de la restauration des données contextuelles, quel est le nombre de données à lire dans la pile STK. Dans l'exemple de réalisation décrit ici à titre d'exemple, le registre de code condition CCR du microprocesseur MP1 de première génération contient deux emplacements disponibles. L'un de ces emplacements est utilisé dans le microprocesseur MP2 pour stocker un drapeau IEA dont la valeur 1 ou 0 forme l'information susmentionnée.

La figure 3 est un organigramme intitulé "Interrupt Handling Sequence" (séquence de traitement d'une interruption) décrivant des opérations de sauvegarde (instructions "PUSH") de données contextuelles dans la pile STK, exécutées par le CPU.

Le CPU sauvegarde tout d'abord le registre PCL et le registre PCH au cours de deux étapes 210 et 220. A l'étape 230 suivante, le CPU teste le contenu du registre PCE.

Si le contenu du registre PCE est nul, le CPU met le drapeau IEA à 1 dans le registre de code condition CCR au cours d'une étape 240. Le CPU exécute ensuite des étapes 250, 260, 270 au cours desquelles il sauvegarde successivement les registres d'index X, d'accumulateur ACC puis de code condition CCR (contenant le drapeau IEA à 1), sans sauvegarder le registre PCE. Le CPU exécute ensuite une étape 280 "JUMP ITSV" ("Jump to Interrupt Service Routine") de saut à une adresse d'exécution d'un sous-programme de traitement de l'interruption, fournie par un vecteur d'interruption.

Si au contraire le contenu du registre PCE n'est pas nul, le CPU met à 0 le drapeau IEA et exécute des étapes 1240, 1250, 1260, 1270 au cours desquelles il sauvegarde successivement les registres PCE, X, ACC et CCR (contenant le drapeau IEA à 0). Le CPU exécute

ensuite l'étape 280 "JUMP ITSV" de saut au sous-programme d'interruption.

Ainsi, le nombre de données contextuelles sauvegardées dans la pile est limité aux cinq registres  
5 PCL, PCH, X, ACC et CCR lorsque le registre PCE est nul.

La figure 4 est un organigramme intitulé "Return from Interrupt " (retour de l'interruption) représentant les opérations de restauration des données contextuelles après exécution du sous-programme d'interruption.

10 Le CPU procède à une récupération des données (instruction "POP") dans l'ordre inverse de leur sauvegarde, selon le principe d'empilement/dépilement LIFO, par déstockage des données contextuelles de la pile et restauration des registres en commençant par les  
15 dernières données enregistrées dans la pile.

Au cours d'étapes 310, 320 et 330, le CPU restaure tout d'abord les registres CCR, ACC et X, successivement. A une étape suivante 340, le CPU teste la valeur du drapeau IEA dans le registre CCR, qui a été  
20 restauré à l'étape 310.

Si le drapeau IEA est à 1, le CPU met à zéro le registre d'adressage programme étendu PCE au cours d'une étape 350, puis restaure les registres PCH et PCL du compteur ordinal PC au cours d'étapes 360, 370. Le CPU  
25 exécute ensuite une étape 380 "JUMP PC" au cours de laquelle il saute à l'adresse contenue dans le compteur ordinal PC tel que restauré, pour reprendre l'exécution du programme initial.

Si au contraire le drapeau IEA est à 0, le CPU  
30 restaure les trois registres PCE, PCH et PCL du compteur ordinal au cours d'étapes 1350, 1360, 1370, puis va à l'étape de saut 380.

L'étape de test 340 peut être une étape implicite si le CPU est une machine d'état à logique câblée. Dans  
35 ce cas, le branchement de l'étape 330 aux étapes 350 ou 1350 est fonction du drapeau IEA utilisé en tant que bit

agissant sur les transitions d'un état à un autre de la machine d'état.

Ainsi, tous les registres de l'unité centrale CCR, ACC, X, PCH, PCL sans oublier le registre d'adressage étendu PCE et le pointeur de pile SPH, SPL sont  
5 entièrement rétablis dans leur état initial.

Le procédé de gestion de pile selon l'invention permet ainsi d'assurer la compatibilité du microprocesseur MP2 avec les programmes développés sur  
10 la première génération de microprocesseur MP1. Avec de tels programmes, le contenu du registre PCE est toujours nul et le registre PCE n'est jamais sauvegardé. Le nombre maximal d'imbrications de sous-programmes est ainsi conservé et les risques de débordement de la pile  
15 sont écartés.

Le registre PCE n'est également pas sauvegardé quand son contenu est nul lors de l'exécution d'un programme élaboré au moyen du nouveau jeu d'instructions, ce qui permet alors d'optimiser le  
20 remplissage de la pile, de réduire le nombre d'opérations effectuées par le CPU pour la sauvegarde et la restauration du contexte, de gagner quelques cycles d'opération et d'accélérer le traitement des interruptions.

La présente invention est ainsi susceptible de  
25 diverses applications, indépendamment du problème de compatibilité entre microprocesseurs. De façon générale, le fait que le registre PCE ne soit pas sauvegardé lorsque son contenu est nul permet d'économiser l'espace disponible dans la pile et d'augmenter le nombre  
30 d'imbrications de sous-programmes. Ce mécanisme de sauvegarde conditionnelle d'un registre peut par ailleurs être étendu à d'autres registres que le registre PCE.

35 Ainsi, une généralisation du procédé qui vient d'être décrit comprend la sauvegarde d'un nombre variable de registres en fonction de la valeur d'un



drapeau stocké dans l'un des registres à sauvegarder, comme ceci apparaîtra à la lumière de la description suivante d'un second exemple de réalisation du procédé selon l'invention.

5        La figure 5 représente un microprocesseur MP3 comprenant, comme le précédent, une unité centrale CPU et un plan mémoire MEM3 dans lequel une zone de taille fixe forme la pile STK du CPU. Le CPU comprend ici un nombre quelconque de registres CR, R1, R2, ..., R<sub>M-1</sub>,  
10 R<sub>M</sub>, ..., R<sub>M+N-1</sub>, par exemple des registres de 8 bits, contenant des données contextuelles. Le CPU comprend également des registres SPH, SPL contenant les deux octets du pointeur de pile SP. Parmi les registres de données contextuelles, les registres CR, R1 à R<sub>M-1</sub>  
15 contiennent des données contextuelles de type essentiel et les registres R<sub>M</sub> à R<sub>M+N-1</sub> contiennent des données contextuelles de type non essentiel, susceptibles de n'être pas utilisées par un programme application. D'autre part un registre particulier, ici le registre  
20 CR, est dédié au stockage d'un drapeau ISV accessible en lecture et en écriture à l'utilisateur du microprocesseur, c'est-à-dire accessible au programme application développé par l'utilisateur.

La valeur du drapeau ISV est contrôlée par le  
25 programme d'application pour faire varier le nombre de registres sauvegardés lors du basculement dans un sous-programme.

La figure 6 est un organigramme intitulé "Interrupt Handling" représentant des étapes de sauvegarde de  
30 données contextuelles exécutées par le CPU en réponse à un signal d'interruption.

Au cours d'une étape 510, le CPU teste l'état du drapeau ISV contenu dans le registre CR.

Si le drapeau ISV est à 0, le CPU exécute une série  
35 520 d'étapes de sauvegarde dans la pile STK du contenu de M registres, ici les registres R1, R2, ..., R<sub>M-1</sub>, CR, le registre CR étant sauvegardé en dernier. Ainsi, M octets

de données contextuelles sont sauvegardés et le pointeur de pile est incrémenté par M. Le CPU exécute ensuite une étape 540 "JUMP ITSV" ("Jump to Interrupt Service Routine") de saut à une adresse fournie par un vecteur  
5 d'interruption, pour l'exécution d'un sous-programme de traitement de l'interruption.

Si au contraire le drapeau ISV est à 1, le CPU exécute une série 530 d'étapes de sauvegarde dans la pile STK de la totalité des M+N registres R1, R2, ... R<sub>M-1</sub>,  
10 R<sub>M</sub>, ... R<sub>M+N-1</sub>, CR, le registre CR étant sauvegardé en dernier. Ainsi, M+N octets de données contextuelles sont sauvegardés et le pointeur de pile est incrémenté par M+N. Le CPU va à l'étape de saut 540, pour exécuter le sous-programme d'interruption.

15 La figure 7 est un organigramme intitulé "Return from Interrupt" illustrant les étapes de restauration des registres du CPU lors du retour au programme initial.

Comme précédemment, les derniers registres  
20 sauvegardés dans la pile sont les premiers à être restaurés.

Ainsi, le CPU restaure tout d'abord le registre CR au cours d'une étape 610. Le CPU teste ensuite au cours d'une étape 620 le drapeau ISV présent dans le registre  
25 restauré.

Si le drapeau ISV est à 0, le CPU exécute une série 630 d'étapes de restauration des registres R<sub>M-1</sub>, ... R2, R1, en déchargeant de la pile les M-1 octets suivants. Ensuite, au cours d'une étape 650 "JUMP PC", le CPU  
30 saute à l'adresse contenue dans le compteur ordinal PC pour reprendre l'exécution du programme initial.

Si le drapeau ISV est à 1, le CPU exécute une série 640 d'étapes de restauration des registres R<sub>M+N+1</sub>, ... R<sub>M</sub>, R<sub>M-1</sub>, ... R2, R1, en déchargeant de la pile les M+N-1 octets  
35 suivants, puis va à l'étape de saut 650.

L'étape de test 620 peut, ici également, être implicite si le CPU est une machine d'état à logique

câblée. Le branchement de l'étape 610 à l'étape 630 ou à l'étape 640 est alors déterminé par le drapeau ISV en tant que bit agissant sur les transitions d'états de la machine d'état.

5 Il apparaîtra clairement à l'homme de l'art que la présente invention est susceptible de diverses autres variantes. La sauvegarde des registres peut notamment être réalisée dans un ordre différent, par exemple de la manière suivante :

- 10 - si  $ISV=0$ , sauvegarde du registre CR puis sauvegarde des registres  $R_1$  à  $R_{M-1}$ , ou  
- si  $ISV=1$ , sauvegarde des registres  $R_M$  à  $R_{M+N-1}$ , puis sauvegarde du registre CR, puis sauvegarde des registres  $R_1$  à  $R_{M-1}$ .

15 Puis, au moment de la restauration :

- restauration des registres  $R_1$  à  $R_{M-1}$ ,
- restauration du registre CR,
- si  $ISV=1$ , restauration des registres  $R_M$  à  $R_{M+N-1}$ .

20 La sauvegarde et la restauration conditionnelles selon l'invention peuvent également concerner plusieurs groupes de registres si un indicateur à plusieurs drapeaux est prévu. Par exemple, un indicateur à deux drapeaux ISF1, ISF2 permet de définir 4 configurations différentes de sauvegarde, et de définir quatre groupes  
25 différents de registres à sauvegarder, les registres contenant des données contextuelles essentielles étant compris dans chacun des groupes de registres.

La présente invention est également susceptible de diverses applications. Ainsi, le terme microprocesseur  
30 ne présente aucun caractère limitatif, la présente invention s'appliquant de façon générale à tout type de circuit intégré comportant une unité centrale pour exécuter des programmes, et comprenant des données contextuelles à sauvegarder lors du basculement d'un  
35 programme à un autre, tels les microcontrôleurs, les DSP (processeurs de traitement de signal), etc..

La mise en œuvre du procédé de l'invention est ainsi particulièrement avantageuse dans les microcontrôleurs qui, en raison des divers périphériques qu'ils comportent, doivent traiter des files  
5 d'interruptions en attente de grande longueur, nécessitant un degré d'imbrication élevé des sous-programmes d'interruption. La présente invention permet alors d'optimiser la gestion de la pile afin de maximaliser le degré d'imbrication.

## REVENDECATIONS

1. Microprocesseur (MP2, MP3) comprenant une unité centrale (CPU) et un plan mémoire (MEM2, MEM3), l'unité centrale comprenant des registres (CCR-PCL...CR- $R_{M+N-1}$ ) contenant des données contextuelles (CTX) et un pointeur de pile (SP, SPH, SPL), le plan mémoire comprenant une pile (STK) pour la sauvegarde de données contextuelles (CTX1-CTXj), l'unité centrale étant agencée pour sauvegarder des données contextuelles lors d'un basculement d'un premier à un second programme,
- 5
- 10 caractérisé en ce que l'unité centrale est agencée pour sauvegarder des données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur d'au moins un drapeau (IEA, ISV) stocké dans un registre (CCR, CR) à sauvegarder.
- 15
2. Microprocesseur selon la revendication 1, dans lequel l'unité centrale (CPU) est agencée pour modifier la valeur du drapeau (IEA) en fonction du contenu d'un registre (PCE), avant de sauvegarder des données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur du drapeau (IEA).
- 20
3. Microprocesseur selon la revendication 2, dans lequel l'unité centrale est agencée pour modifier la valeur du drapeau (IEA) en fonction du contenu d'un registre d'adressage étendu (PCE) d'un compteur ordinal (PC) de l'unité centrale.
- 25
4. Microprocesseur selon la revendication 3, dans lequel l'unité centrale est agencée pour :
- 30
- quand le contenu du registre d'adressage étendu est égal à 0, sauvegarder tous les registres de l'unité centrale contenant des données contextuelles à l'exception du registre d'adressage étendu (PCE), ou,
- 35



- quand le contenu du registre d'adressage étendu n'est pas égal à 0, sauvegarder tous les registres de l'unité centrale contenant des données contextuelles y compris le registre d'adressage étendu (PCE).

5

5. Microprocesseur selon l'une des revendications 1 à 4, dans lequel l'unité centrale est agencée pour effectuer un test sur la valeur du drapeau (ISV, IEA) afin de déterminer le nombre de registres à sauvegarder.

10

6. Microprocesseur selon l'une des revendications 1 à 5, dans lequel l'unité centrale est agencée pour, lors du retour au premier programme :

- restaurer le registre (CCR, CR) contenant le drapeau (IEA, ISV) et,
- restaurer des données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur du drapeau (IEA, ISV) présent dans le registre restauré.

20

7. Microprocesseur selon l'une des revendications 1 à 6, dans lequel l'unité centrale est agencée pour sauvegarder en dernier le registre (CCR, CR) contenant le drapeau (IEA, ISV).

25

8. Microprocesseur selon l'une des revendications 1 à 7, dans lequel le drapeau (IEA) comprend au moins un bit d'un registre (CCR) contenant des drapeaux de code condition.

30

9. Procédé de gestion de la pile (STK) d'un microprocesseur (MP2, MP3) comprenant une unité centrale (CPU) et un plan mémoire (MEM2, MEM3), l'unité centrale comprenant des registres (CCR-PCL...CR- $R_{M+N-1}$ ) contenant des données contextuelles (CTX) et un pointeur de pile (SP), la pile (STK) étant une zone du plan mémoire

35

dédiée à la sauvegarde de données contextuelles lors d'un basculement d'un premier à un second programme,

caractérisé en ce qu'il comprend la sauvegarde de données contextuelles contenues dans un nombre variable  
5 de registres qui est fonction de la valeur d'au moins un drapeau (IEA, ISV) stocké dans un registre (CCR, CR) à sauvegarder.

10. Procédé selon la revendication 9, comprenant  
10 une étape de modification de la valeur du drapeau (IEA) en fonction du contenu d'un registre (PCE), avant la sauvegarde de données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur du drapeau (IEA).

15

11. Procédé selon la revendication 10, dans lequel la valeur du drapeau (IEA) est modifiée en fonction du contenu d'un registre d'adressage étendu (PCE) d'un compteur ordinal (PC) de l'unité centrale.

20

12. Procédé selon la revendication 11, comprenant les étapes suivantes :

- quand le contenu du registre d'adressage étendu est égal à 0, sauvegarder tous les registres de l'unité  
25 centrale contenant des données contextuelles à l'exception du registre d'adressage étendu (PCE), ou,  
- quand le contenu du registre d'adressage étendu n'est pas égal à 0, sauvegarder tous les registres de l'unité centrale contenant des données contextuelles y compris  
30 le registre d'adressage étendu (PCE).

13. Procédé selon l'une des revendications 9 à 12, comprenant une étape de test de la valeur du drapeau (ISV, IEA) pour déterminer le nombre de registres  
35 contenant les données à sauvegarder.



14. Procédé selon l'une des revendications 9 à 13, comprenant les étapes suivantes:

- restauration du registre (CCR, CR) contenant le drapeau (IEA, ISV), puis,

5 - restauration de données contextuelles contenues dans un nombre variable de registres qui est fonction de la valeur du drapeau (IEA, ISV) présent dans le registre restauré.

10 15. Procédé selon l'une des revendications 9 à 14, dans lequel le registre (CCR, CR) contenant le drapeau (IEA, ISV) est sauvegardé le dernier et est restauré le premier.

15 16. Procédé selon l'une des revendications 9 à 15, dans lequel le drapeau (IEA) est formé par au moins un bit d'un registre (CCR) contenant des drapeaux de code condition.



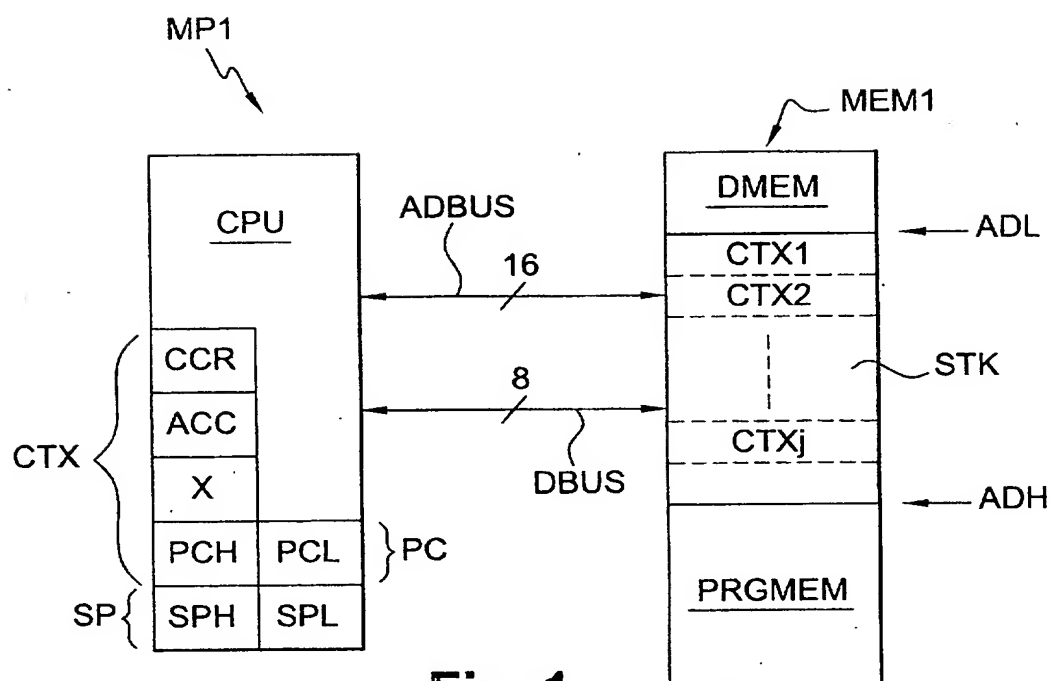


Fig. 1

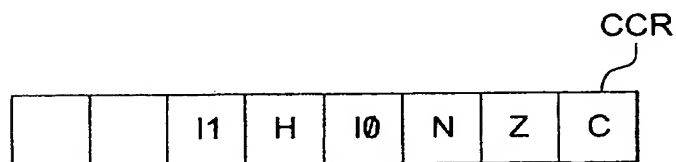
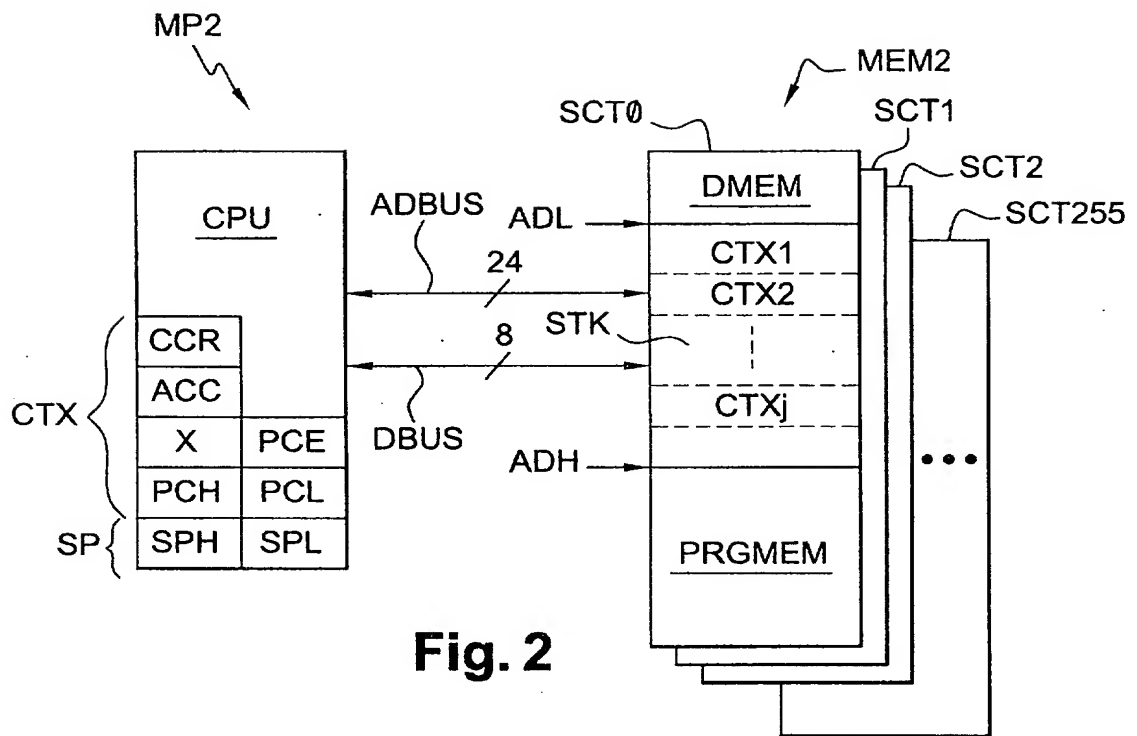


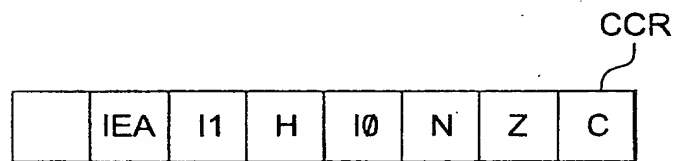
Fig. 1A



2 / 7

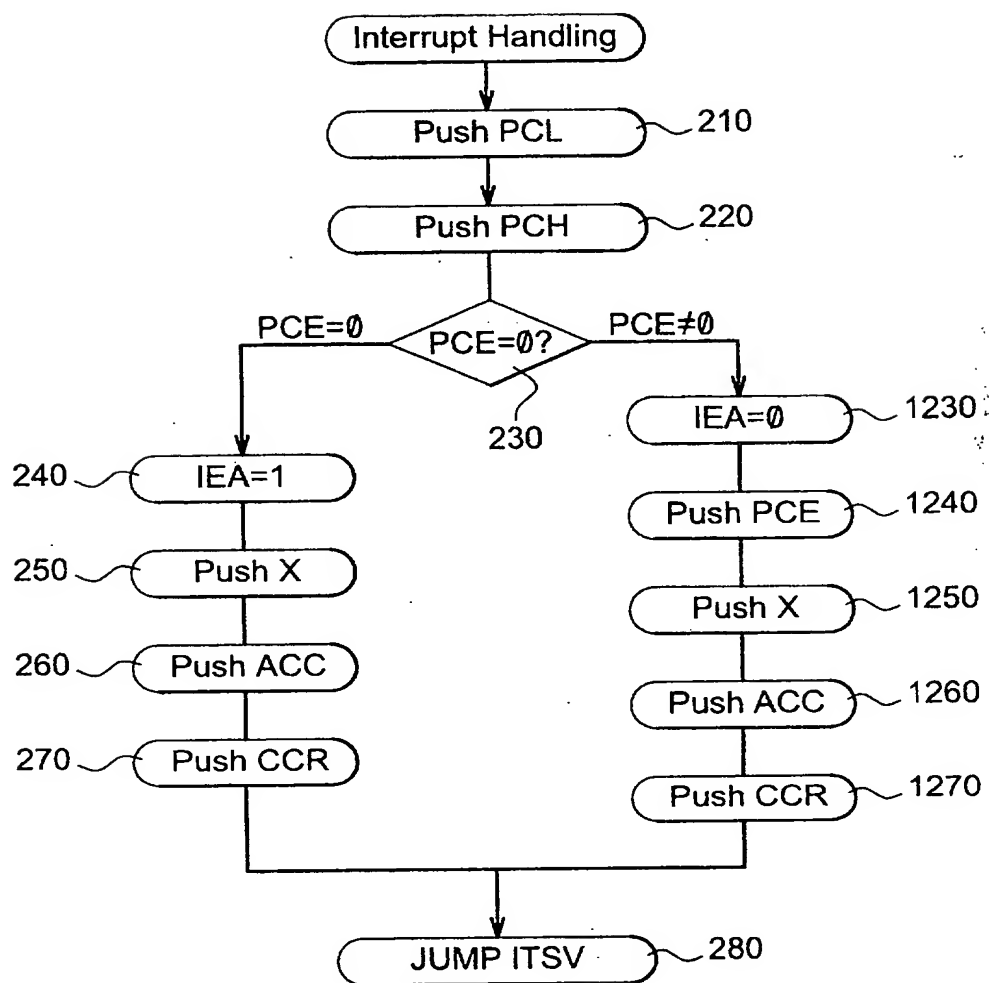


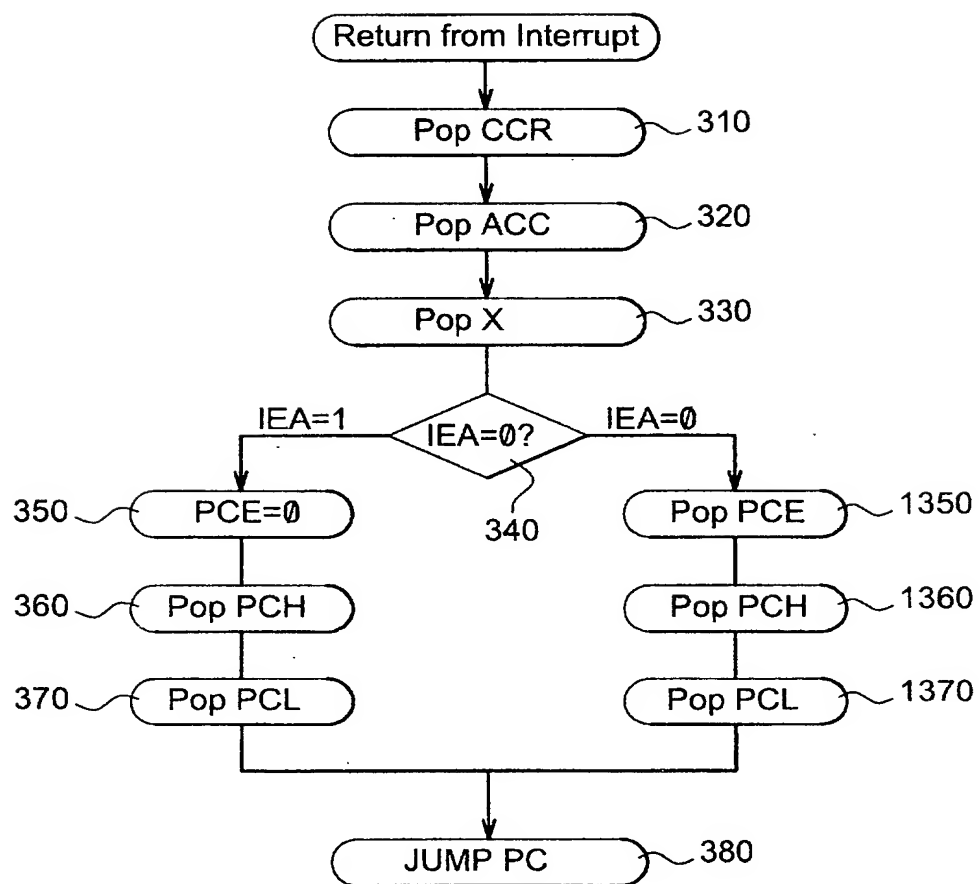
**Fig. 2**

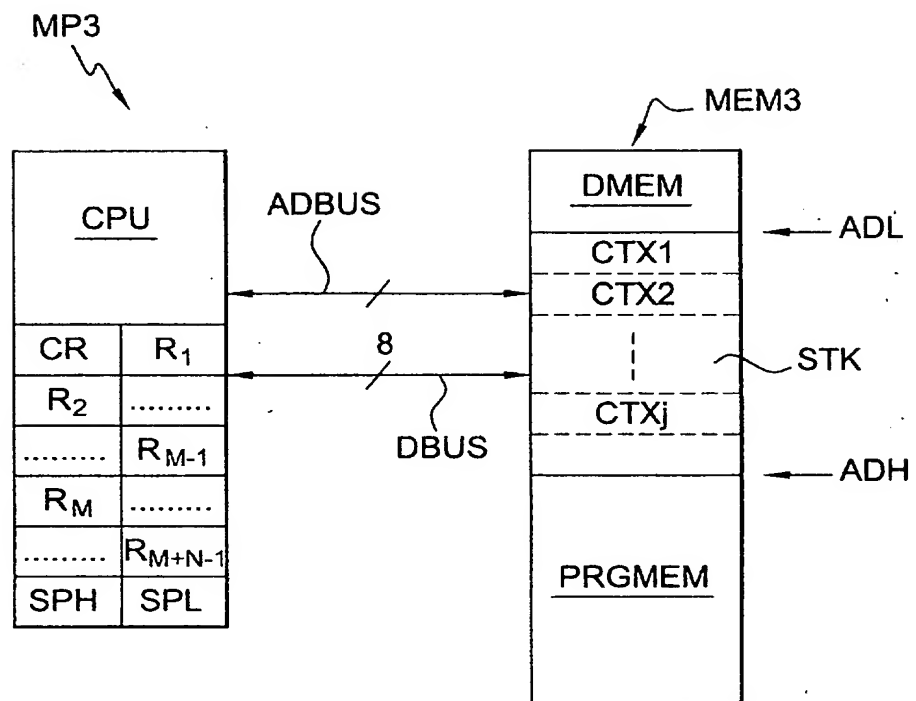


**Fig. 2A**

3/7

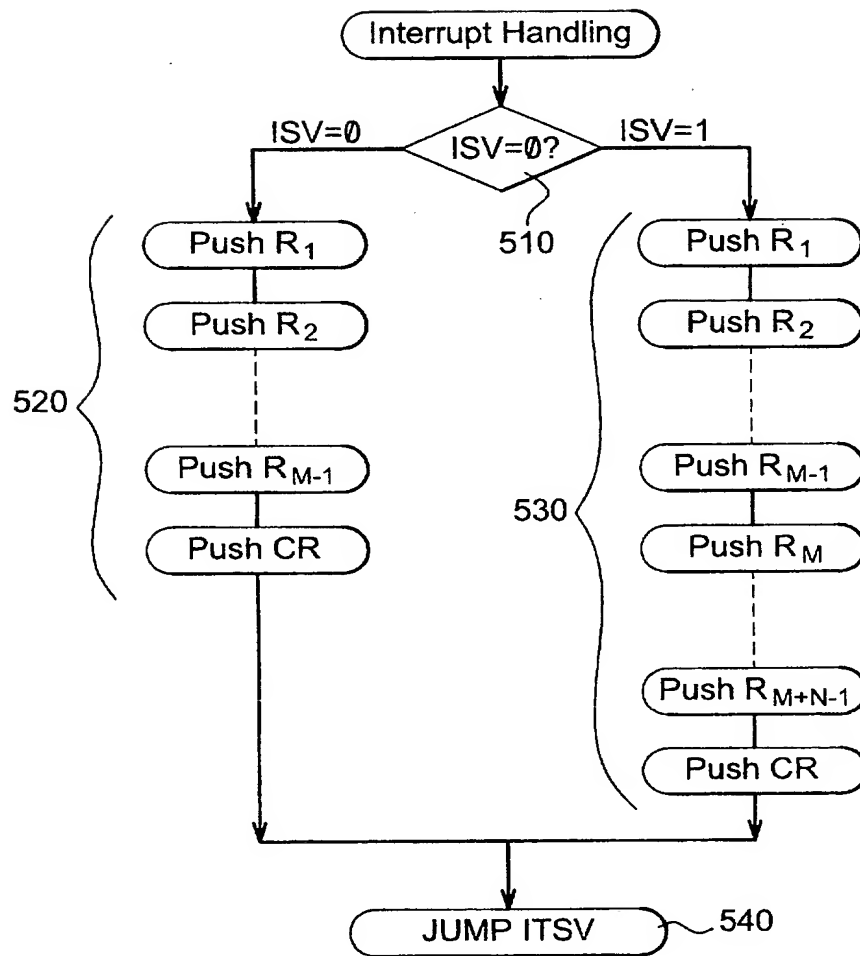
**Fig. 3**

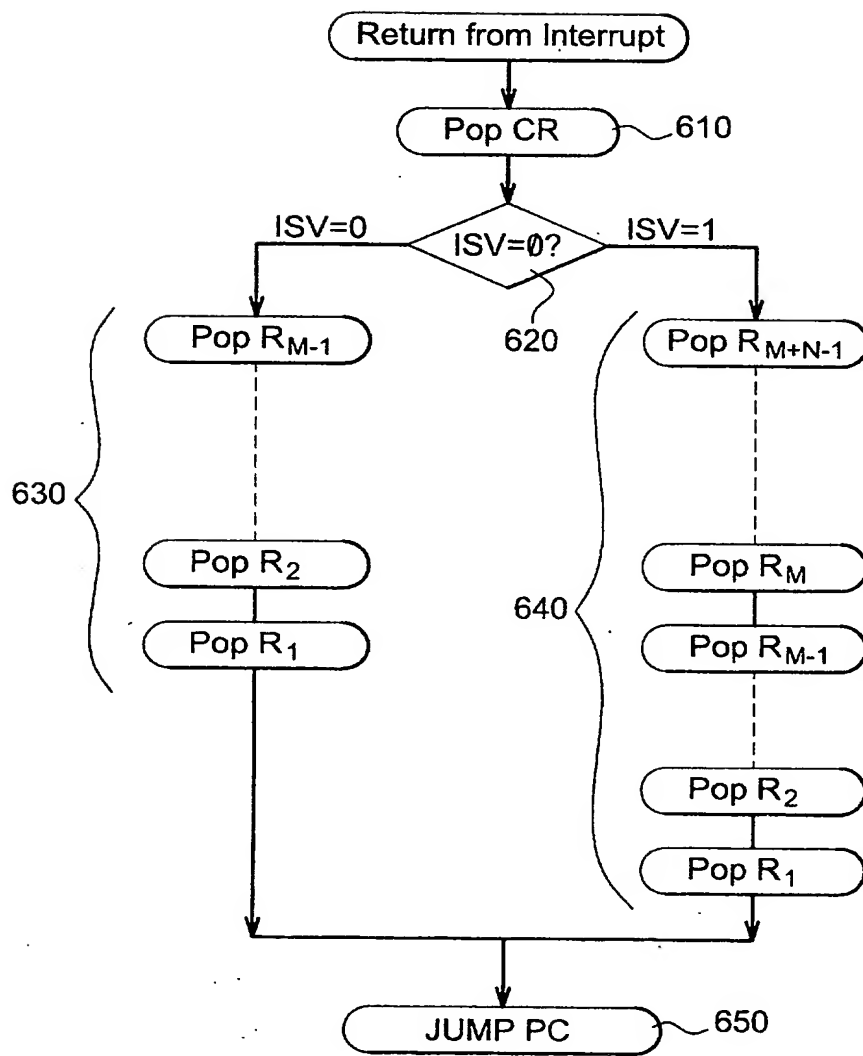
**Fig. 4**

**Fig. 5**



6 / 7

**Fig. 6**

**Fig. 7**

**BREVET D'INVENTION****CERTIFICAT D'UTILITÉ**

Code de la propriété intellectuelle - Livre VI



N° 11 235\*02

## DÉPARTEMENT DES BREVETS

26 bis, rue de Saint Pétersbourg

75800 Paris Cedex 08

Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

DÉSIGNATION D'INVENTEUR(S) Page N° 1.. / 1..

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 W / 260899

17 FEV 2003 Vos références pour ce dossier (facultatif) IS INPT MARSEILLE		100213 FR	
N° D'ENREGISTREMENT NATIONAL 0301879		03 01 879	
<b>TITRE DE L'INVENTION (200 caractères ou espaces maximum)</b> Procédé de gestion d'une pile de microprocesseur pour la sauvegarde de données contextuelles			
<b>LE(S) DEMANDEUR(S) :</b> MARCHAND André OMNIPAT 24, Place des Martyrs de la Résistance 13100 AIX EN PROVENCE			
<b>DESIGNE(NT) EN TANT QU'INVENTEUR(S) :</b> (Indiquez en haut à droite «Page N° 1/1» S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).			
Nom		PADMANABHAN	
Prénoms		Gosagan	
Adresse	Rue	C/O OMNIPAT 24 Place des Martyrs de la Résistance	
	Code postal et ville	13100	AIX EN PROVENCE
Société d'appartenance (facultatif)			
Nom		DAVIDESCU	
Prénoms		Dragos	
Adresse	Rue	C/O OMNIPAT 24 Place des Martyrs de la Résistance	
	Code postal et ville	13100	AIX EN PROVENCE
Société d'appartenance (facultatif)			
Nom		ROCHE	
Prénoms		Franck	
Adresse	Rue	C/O OMNIPAT 24 Place des Martyrs de la Résistance	
	Code postal et ville	13100	AIX EN PROVENCE
Société d'appartenance (facultatif)			
<b>DATE ET SIGNATURE(S)</b> <b>DU (DES) DEMANDEUR(S)</b> <b>OU DU MANDATAIRE</b> (Nom et qualité du signataire) Aix en Provence, le 11 février 2003 MARCHAND André - CPI N° 95 0303 OMNIPAT			